CLAIMS

What is claimed is:

1. A computing system, comprising:

a rounding apparatus to accepts an input value that is a real number represented in floating-point format, and to perform a rounding operation on the input value to generate an output value that is an integer represented in floating-point format;

a memory to store a computer program that utilizes the rounding apparatus; and

a central processing unit (CPU) to execute the computer program, the CPU is cooperatively connected to the rounding apparatus and the memory.

2. The system of claim 1, wherein the rounding apparatus uses a truncation technique to round the input value.

3. The system of claim 2, wherein the rounding apparatus includes:

a floating-point to integer converter to truncate the input value to convert the input value to an integer represented in an integer format; and

an integer to floating-point converter to convert the integer represented in an integer format to the output value.

4. The system of claim 1, wherein the rounding apparatus rounds the input value to the nearest integer.

5. The system of claim 4, wherein the rounding apparatus includes:

an "AND" operator to extract a sign bit of the input value;

an "OR" operator to generate an adjustment value based on the sign bit;

an ADD operator to compute an adjusted input value by adding the adjustment value to the input value, the adjusted input value is a real number represented in floating-point format;

a floating-point to integer converter to truncate a fractional portion of the adjusted input value to convert the adjusted input value to an integer represented in an integer format; and

an integer to floating-point converter to convert the integer represented in an integer format to generate the output value.

6. The system of claim 5, wherein the "AND" operator extracts the sign bit of the input value by performing a bit-wise logical AND operation on the input value and a sign mask.

7. The system of claim 5, wherein the "OR" operator generates the adjustment value by performing a bit-wise logical OR operation on the sign bit and a real value of 0.5.

8. The system of claim 1, wherein the rounding apparatus rounds the input value toward minus infinity (-∞).

9. The system of claim 8, wherein the rounding apparatus includes:

a floating-point to integer converter to truncate an input value to convert the input value to a first integer represented in an integer format;

an integer to floating-point converter to convert the first integer represented in an integer format to a second integer represented in floating-point format;

a first SUBTRACT operator to compute a fractional portion of the input value using the second integer;

a "less than" comparator to generate a boolean mask based on the fractional portion of the input value;

an "AND" operator to use the boolean mask to generate an adjustment value represented in floating-point format; and

a second SUBTRACT operator to subtract the adjustment value from the input value to generate the output value.

10. The system of claim 9, wherein the first SUBTRACT operator computes the fractional portion of the input value by subtracting the second integer from the input value.

11. The system of claim 9, wherein the "less than" comparator generates the boolean mask by comparing the fractional portion of the input value to a real value of 0.0.

12. The system of claim 9, wherein the "AND" operator generates the adjustment value by performing a bit-wise logical AND operation on the boolean mask and a real value of 1.0.

13. The system of claim 1, wherein the rounding apparatus rounds the input value toward plus infinity (+∞).

14. The system of claim 13, wherein the rounding apparatus includes:

a floating-point to integer converter to truncate an input value to convert the input value to a first integer represented in an integer format;

an integer to floating-point converter to convert the first integer represented in an integer format to a second integer represented in floating-point format;

Sub a1 a SUBTRACT operator to compute a fractional portion of the input value using the second integer;

a "greater-than" comparator to generate a boolean mask based on the fractional portion of the input value;

an "AND" operator to use the boolean mask to generate an adjustment value represented in floating-point format; and

an ADD operator to add the adjustment value to the input value to generate the output value.

15. The system of claim 14, wherein the SUBTRACT operator computes the fractional portion of the input value by subtracting the second integer from the input value.

16. The system of claim 14, wherein the "greater-than" comparator generates the boolean mask by comparing the fractional portion of the input value to a real value of 0.0.

17. The system of claim 14, wherein the "AND" operator generates the adjustment value by performing a bit-wise logical AND operation on the boolean mask and a real value of 1.0.

18. A method comprising:

accepting an input value that is a real number represented in floating-point format;

converting the input value to a first integer;

converting the first integer represented to a second integer; and

storing the second integer as an output value.

Sub a1

19. The method of claim 18, wherein converting the input value to a first integer comprises:

representing the first integer in an integer format.

20. The method of claim 18, wherein converting the first integer to the second integer comprises:

representing the second integer in floating-point format.

21. A method comprising:

building an adjustment value represented in floating-point format;

adding the adjustment value to an input value to generate an adjusted input value represented in floating-point format;

truncating the adjusted input value to convert the adjusted input value to a first integer represented in an integer format;

converting the first integer represented in an integer format to a second integer represented in floating-point format; and

storing the second integer as an output value.

22. The method of claim 21, wherein building the adjustment value comprises:

extracting a sign bit of the input value by performing a bit-wise logical AND operation on the input value and a sign mask.

23. The method of claim 21, wherein building the adjustment value comprises:

building the adjustment value by performing a bit-wise logical OR operation on a real value of 0.5 and a sign bit extracted from the input value.

Sub a4

24. A method comprising:

generating a first integer represented in an integer format by truncating an input value;

converting the first integer represented in an integer format to a second integer represented in floating-point format;

computing a fractional portion of the input value using the second integer represented in floating-point format;

generating a boolean value using the fractional portion of the input value;

creating an adjustment value using the boolean value;

computing a rounded input value by subtracting the adjustment value from the input value.

25. The method of claim 24, wherein computing the fractional portion of the input value comprises:

subtracting the second integer represented in floating-point format from the input value to generate the fractional portion of the input value.

26. The method of claim 24, wherein generating the boolean value comprises comparing the fractional portion of the input value to a real value of 0.0.

27. The method of claim 24, wherein creating an adjustment value comprises performing a bit-wise logical AND operation on the boolean value and a real value of 1.0.

28. A method comprising:

generating a first integer represented in an integer format by truncating an input value;

Sub a1

~~converting the first integer represented in an integer format to a second integer represented in floating-point format;~~

~~subtracting the second integer represented in floating-point format from the input value to generate a fractional portion of the input value;~~

~~generating a boolean value using the fractional portion of the input value;~~

~~creating an adjustment value using the boolean value;~~

~~adding the adjustment value to the input value to generate a rounded input value.~~

~~29. The method of claim 28, wherein creating an adjustment value comprises:~~

~~comparing the fractional portion of the input value to a real value of 0.0.~~

~~30. The method of claim 28, wherein creating an adjustment value comprises:~~

~~performing a bit-wise logical AND operation on the boolean value and a real value of 1.0.~~

~~31. A machine-readable medium comprising instructions which, when executed by a machine, cause the machine to perform operations comprising:~~

~~a first code segment truncates the input value to convert the input value to a first integer; and~~

~~a second code segment integer to convert the first integer to a second integer.~~

~~32. The machine-readable medium of claim 31, wherein the first integer is represented in an integer format.~~

Sub C1

~~33. The machine-readable medium of claim 31, wherein the second integer is represented in floating-point format.~~

~~34. A machine-readable medium comprising instructions which, when executed by a machine, cause the machine to perform operations comprising:~~

~~a first code segment to extract a sign bit of the input value;~~

~~a second code segment to generate an adjustment value based on the sign bit;~~

~~a third code segment to compute an adjusted input value represented in floating-point format;~~

~~a fourth code segment to truncate a fractional portion of the adjusted input value to convert the adjusted input value to an integer represented in an integer format; and~~

~~a fifth code segment to convert the integer represented in an integer format to generate the output value.~~

~~35. The machine-readable medium of claim 34, wherein the second code segment generates the adjustment value by performing a bit-wise logical OR operation on the sign bit and a value of 0.5.~~

~~36. The machine-readable medium of claim 34, wherein the third code segment computes the adjusted input value by adding the adjustment value to the input value.~~

~~37. A machine-readable medium comprising instructions which, when executed by a machine, cause the machine to perform operations comprising:~~

~~a first code segment to truncate an input value to convert the input value to a first integer represented in an integer format;~~

~~a second code segment to floating-point converter to convert the first integer represented in an integer format to a second integer represented in floating-point format;~~

~~a third code segment to subtract the second integer from the input value to compute a fractional portion;~~

~~a fourth code segment to generate a boolean mask based on the fractional portion of the input value;~~

~~a fifth code segment to perform a bit-wise logical AND operation on the boolean mask and a real value of 1.0 to generate an adjustment value represented in floating-point format; and~~

~~a sixth code segment to subtract the adjustment value from the input value to generate the output value represented in floating-point format.~~

~~38. The machine-readable medium of claim 37, wherein the fourth code segment generates the boolean mask by comparing the fractional portion of the input value to a real value of 0.0.~~

~~39. The machine-readable medium of claim 37, wherein the fifth code segment generates the adjustment value by performing a bit-wise logical AND operation on the boolean mask and a real value of 1.0.~~

~~40. A machine-readable medium comprising instructions which, when executed by a machine, cause the machine to perform operations comprising:~~

~~a first code segment to truncate an input value to convert the input value to a first integer represented in an integer format;~~

~~a second code segment to floating-point converter to convert the first integer represented in an integer format to a second integer represented in floating-point format;~~

Sub a1

~~a third code segment to subtract the second integer from the input value to compute a fractional portion of the input value;~~

~~a fourth code segment to generate a boolean mask based on the fractional portion of the input value;~~

~~a fifth code segment to an adjustment value represented in floating-point format; and~~

~~a sixth code segment to subtract the adjustment value from the input value to generate the output value represented in floating-point format.~~

~~41. The machine-readable medium of claim 40, wherein the fourth code segment generates the boolean mask by comparing the fractional portion of the input value to a real value of 0.0.~~

~~42. The machine-readable medium of claim 40, wherein the fifth code segment generates the adjustment value by performing a bit-wise logical AND operation on the boolean mask and a real value of 1.0.~~

add a1